

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-232140

(43)Date of publication of application : 27.08.1999

(51)Int.Cl.

G06F 11/30  
G06F 9/46

(21)Application number : 10-043016

(71)Applicant : NEC COMMUN SYST LTD

(22)Date of filing : 10.02.1998

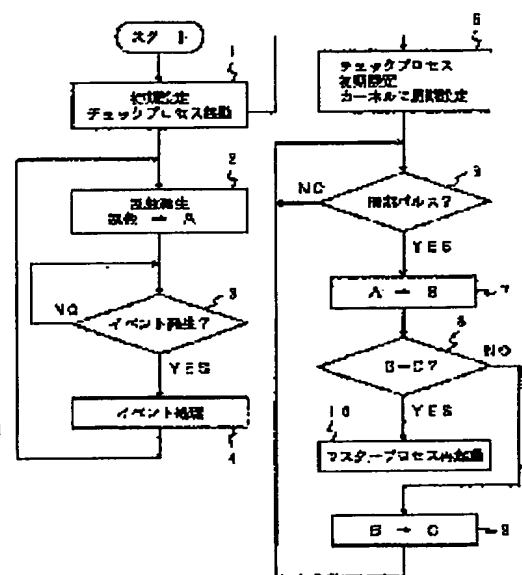
(72)Inventor : MIYAMOTO TAKESHI

## (54) PROCESSING STOPPAGE AVOIDING METHOD

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To make promptly and automatically restartable a master process in the case where a deadlock is generated by detecting whether or not the master process generates a random number for one or more times within a prescribed cycle.

**SOLUTION:** The master process is provided with a random number generation circuit and generates one random number and writes it in a register A (S2) after initialization (S1) and after an event processing (S4). A check process sets a predetermined cycle to a kernel (S5) other than general initialization. Being triggered by pulses, the check process compares the value of observing the random number generated by the master process in the cycle one before with the value of observing it at the present point of time (S7 and S8). The matching of the two indicates that the master process does not execute random number generation during the prescribed cycle and is judged as the hang-up of the master process and the check process reactivates the mater process (S10).



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-232140

(43) 公開日 平成11年(1999) 8月27日

(51) Int.Cl.<sup>9</sup>

G 0 6 F 11/30  
9/46

識別記号

3 0 5  
3 4 0

F I

G 0 6 F 11/30  
9/46

3 0 5 C  
3 4 0 G

審査請求 有 請求項の数 4 F D (全 5 頁)

(21) 出願番号 特願平10-43016

(22) 出願日 平成10年(1998) 2月10日

(71) 出願人 000232254

日本電気通信システム株式会社  
東京都港区三田1丁目4番28号

(72) 発明者 宮元 毅

東京都港区三田1丁目4番28号 日本電気  
通信システム株式会社内

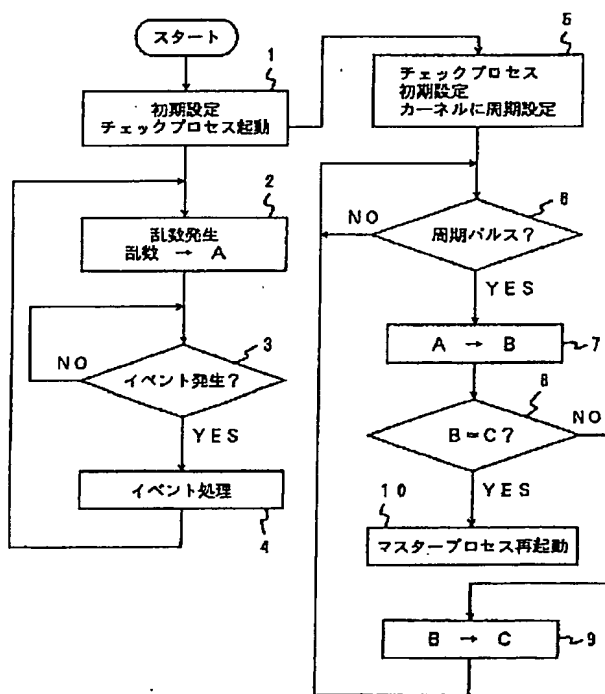
(74) 代理人 弁理士 高橋 友二

(54) 【発明の名称】 処理停止回避方法

(57) 【要約】

【課題】 発生が予期される各イベントに対する処理とそのイベントを処理する子プロセスが予め定められていて、イベントが発生するごとに、その発生したイベントの処理を当該イベントに対応する子プロセスに指示するマスタープロセスのデッドロックを回避する。

【解決手段】 マスタープロセスの実行をチェックするチェックプロセスを設け、マスタープロセスはイベント処理のステップの後で乱数を発生し、チェックプロセスはマスタープロセスが乱数発生を実行していることを検出して、マスタープロセスが動作していることを確認する。



## 【特許請求の範囲】

【請求項 1】 発生が予期される各イベントに対する処理とそのイベントを処理するサブプロセスが予め定められていて、イベントが発生することに、その発生したイベントの処理を当該イベントに対応するサブプロセスに指示するマスタープロセスのデッドロックを回避する処理停止回避方法において、

当該マスタープロセスの初期設定に際し、予め定められたチェックプロセスを起動するステップ 1、

前記マスタープロセスに乱数発生器を備え、ステップ 1 の後イベントの発生を待つ間に一つの乱数を発生して、これをレジスタ A に書き込むステップ 2、

イベントが発生したときそのイベントの処理に対応するサブプロセスに指示した後ステップ 2 に戻るステップ 4、

前記チェックプロセスの初期設定に際し、予め定められた周期をカーネルに設定するステップ 5、

この初期設定の後、前記チェックプロセスで前記予め定められた周期のパルスの到来を待つステップ 6、

前記予め定められた周期のパルスが到来するとチェックプロセスで前記レジスタ A の内容をレジスタ B にロードするステップ 7、

このステップ 7 の後、レジスタ B の内容をレジスタ C の内容と比較するステップ 8、

このステップ 8 の比較で、レジスタ B の内容とレジスタ C の内容とに差異があるとき、レジスタ B の内容をレジスタ C にロードしてステップ 6 に戻るステップ 9、

前記ステップ 8 の比較で、レジスタ B の内容とレジスタ C の内容とが一致するとき、前記マスタープロセスを再起動するプロセス 1 0、

を備えた処理停止回避方法。

【請求項 2】 請求項 1 記載の処理停止回避方法において、前記ステップ 5 で設定される周期は、前記発生を予期されるイベント中、最大の処理時間を必要とするイベントの処理時間の 2 倍程度に定められることを特徴とする処理停止回避方法。

【請求項 3】 請求項 1 記載の処理停止回避方法において、前記チェックプロセスは更にステップ 7 とステップ 8 との間に一つの乱数を発生してこれをレジスタ D に書き込むステップ 1 9 を備え、

前記マスタープロセスは更に、初期設定に際し、予め定められたチェックプロセスを起動し、かつカウンタに所定数値をリセットするステップ 1 1 と、

ステップ 4 の後ステップ 2 へ戻る前に、レジスタ D の内容をレジスタ E にロードするステップ 1 2 と、

このステップ 1 2 の後、レジスタ E の内容をレジスタ F の内容と比較するステップ 1 3 と、

このステップ 1 3 の比較で、レジスタ E の内容とレジスタ F の内容とに差異があるとき、レジスタ E の内容をレジスタ F にロードし前記カウンタをプリセットしてステップ 2 に戻るステップ 1 4 と、

前記ステップ 1 3 の比較で、レジスタ E の内容とレジスタ F の内容とが一致するとき、前記カウンタの数値を検査するステップ 1 5 と、

検査したカウンタの数値が正であれば、カウンタの数値を 1 だけ減少してステップ 2 に戻るステップ 1 6 と、

検査したカウンタの数値が 0 又は負であれば、前記チェックプロセスを再起動し、前記カウンタをプリセットしてステップ 2 に戻るステップ 1 8 と、

を備えたことを特徴とする処理停止回避方法。

【請求項 4】 請求項 3 記載の処理停止回避方法において前記ステップ 1 1 又は前記ステップ 1 8 で前記カウンタにプリセットされる数値は、前記ステップ 5 により設定される周期と、発生が予期される各イベントの処理時間とを考慮して定められることを特徴とする処理停止回避方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、マスタープロセスがデッドロックすることを防止する処理停止回避方法に関するものである。たとえば、システム内において各種の通信制御を行う場合、所定のイベントが発生し、その発生したイベントの処理として通信制御が行われるが、発生するイベントとそのイベント処理として行われる通信制御と、この通信制御を行うサブプロセスとは予め定められていて、発生した各イベントを識別し、そのイベントを処理するように予め定められている各サブプロセスを制御、管理するプロセスをマスタープロセスと称しているが、本発明はこのようなマスタープロセスの処理停止回避方法に関するものである。

## 【0002】

【従来の技術】電子計算機の暴走（ハングアップを含む）を防止する方法については、従来色々の方法が開示されている。然しながら、従来開示されている暴走防止方法は、主として、制御プログラムが内蔵されていて、この制御プログラムを読み出すプログラムカウンタを有する電子計算機に関するものであって、上述のマスタープロセスはデッドロックしないような簡潔な制御処理として、処理停止回避方法は講じられてなかった。

【0003】図 3 は従来のマスタープロセスの処理の流れを示すフローチャートで、101 は初期設定のステップ、3 はイベントの発生を待っているステップ、4 はイベントが発生したとき、発生したイベントを処理するイベント処理ステップである。発生したイベントがどのイベントであるかは容易に識別でき、そのイベントに対しどのサブプロセスを起動すべきかということも予め定められているので、マスタープロセスの制御処理は簡潔であり、処理停止回避方法は講じられてない。

## 【0004】

【発明が解決しようとする課題】然し、このような簡単なマスタープロセスでもデッドロックすることがあり、

このデッドロックが検出され、マスタープロセスが外部からリセットされるまでには、通常多くの時間が空費され、そのマスタープロセスがシステム内での通信制御のマスタープロセスである場合、その間システム内の通信が途絶えるという事故が発生する。

【0005】従って、本発明の目的は、このようなマスタープロセスにも処理停止回避の方法を講じ、何らかの理由によりデッドロックを生じた場合、速やかにマスタープロセスを自動的に再起動する方法を提供することである。

【0006】

【課題を解決するための手段】本発明では、マスタープロセスがイベントの発生を待っているステップに移る前に乱数を発生し、所定周期内に一回以上の乱数が発生されたか否かを検出することによってマスタープロセスのデッドロックを検出した。

【0007】すなわち、本発明の一実施形態では、マスタープロセスの初期設定に際し、予め定められたチェックプロセスを起動するステップ1、マスタープロセスに乱数発生器を備え、ステップ1の後イベントの発生を待つ間に一つの乱数を発生して、これをレジスタAに書き込むステップ2、イベントが発生したときそのイベントの処理を対応する子プロセスに指示した後ステップ2に戻るステップ4、前記チェックプロセスの初期設定に際し、予め定められた周期をカーネルに設定するステップ5、この初期設定の後、前記チェックプロセスで予め定められた周期のパルスの到来を待つステップ6、予め定められた周期のパルスが到来するとチェックプロセスでレジスタAの内容をレジスタBにロードするステップ7、このステップ7の後、レジスタBの内容をレジスタCの内容と比較するステップ8、このステップ8の比較で、レジスタBの内容とレジスタCの内容とに差異があるとき、レジスタBの内容をレジスタCにロードしてステップ6に戻るステップ9、ステップ8の比較で、レジスタBの内容とレジスタCの内容とが一致するとき、前記マスタープロセスを再起動するプロセス10、を備えた処理停止回避方法を提供する。

【0008】

【発明の実施の形態】以下、図面について、本発明の実施の形態を説明する。図1は本発明の実施の一形態を示すフローチャートで、符号1～10は各処理ステップを示す。ステップ1の初期設定では図3のステップ101に示す初期設定と同様な一般的な初期設定に加えて予め定められたチェックプロセスを起動する。ステップ3、4は図3のステップ3、4と同様なステップで、イベントの発生を待っており、イベントが発生すると、これを処理して次のイベントを待つ。

【0009】この発明のマスタープロセスは乱数発生回路を備えていて、ステップ1の後及びステップ4の後に一つの乱数を発生し、これをレジスタAに書き込む。書

き込みによってレジスタに従来保持されていた内容は消去される。従って、初期設定のステップ1においてレジスタAの内容をリセットして0にしておくのが一般であるが乱数をセットするレジスタであるので、初期設定でリセットすることは、必ずしも必要ではない。レジスタAに関する以上の説明は、後述のレジスタB、C、D、E、Fにも同様に適用される。

【0010】ステップ1によって起動されたチェックプロセスはステップ5で一般的な初期設定の他に、予め定められた周期をカーネルに設定する。チェックプロセスも簡単なプロセスで、制御プログラムの内蔵もなく、プログラムカウンタもない。カーネルがプログラムカウンタに代わり設定周期ごとにパルスを発生する。このパルスにトリガされてチェックプロセスはステップ7、8、9を繰り返す。ステップ7ではレジスタAの内容をレジスタBにロードし（書き込み）、ステップ8ではレジスタBの内容がレジスタCの内容と一致するか否かを検査し、一致しなければ、ステップ9でレジスタBの内容をレジスタCにロードして、ステップ6に戻る。

【0011】すなわち、ステップ8では、マスタープロセスが発生する乱数を一つ前の周期で観察した値と、現在時点で観察した値とを比較することになる。この二つが一致することは、所定の周期の間にマスタープロセスはステップ2を実行しなかったことを意味する。この所定の周期は、ステップ4でマスタープロセスが子プロセスにイベント処理を指示し、次のイベントが発生するまでの時間が最も長いイベントの処理時間の2倍程度に定めてあるので、この周期の間にステップ2が実行されないことは、マスタープロセスがステップ4を実行した後ステップ2を実行しなかったことを示すもので、マスタープロセスのハングアップと判断し、チェックプロセスはステップ10に移り、マスタープロセスを再起動し、ステップ1から実行させる。

【0012】更に、必要があれば、チェックプロセスのハングアップをマスタープロセスでチェックすることもできる。図2は本発明の他の実施形態を示すフローチャートで、図2においてステップ2、3、4、5、6、7、8、9、10は図1の同一符号のステップと同様であるので、図2には簡略化して示してある。

【0013】図2に示す実施形態では、チェックプロセスも乱数発生回路を持ち、ステップ7とステップ8の間に挿入されるステップ19で一つの乱数を発生しレジスタDにロードする。マスタープロセスはステップ4からステップ2へ戻る間に、図2に示すステップ12～18を挿入してチェックプロセスのハングアップを検出する。そのため、ステップ11の初期設定ではステップ1の一般初期設定、チェックプロセス起動の外に、カウンタに所定数値をプリセットする処理を行う。

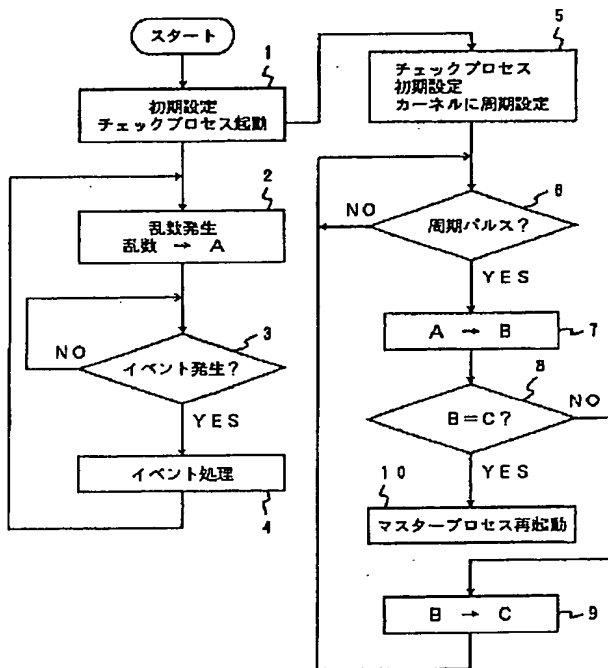
【0014】マスタープロセスはステップ12でレジスタDの内容をレジスタEにロードし、ステップ13でレ

ジスタEの内容をレジスタFの内容と比較し、両内容に差異があればレジスタEの内容をレジスタFにロードしてステップ2に戻る。レジスタEの内容とレジスタFの内容とが一致すると、ステップ19の実行ミスを疑って見る。然し、ステップ19は所定周期毎に実行され、ステップ13はイベントの発生ごとに行われているので、イベントの発生が何回連続すると、その合計時間が必ず所定周期より長くなるという数値が予め定められていて、その数値がカウンタにプリセットされている。

【0015】従って、ステップ15でカウンタの計数値を調べ、それが正のうちはステップ16に移りカウンタの計数値から1を減算して（カウンタデクリメント）ステップ2に戻る。その内ステップ19が実行されると、ステップ13の判定がNOとなって、ステップ14に移り、ステップ18でカウンタをプリセットした後ステップ2に戻る。ステップ13の判定が連続して所定回数分YESとなると、カウンタの計数値は0となり、ステップ15でチェックプロセスのハングアップと判断され、ステップ17でチェックプロセス再起動の上、ステップ18でカウンタをリセットしてステップ2に戻る。

【0016】

【図1】



\* 【発明の効果】 以上のように、本発明によれば、マスタープロセスのデッドロックを簡単なチェックプロセスで検出して自動的に再起動することができ、必要があれば、更に当該チェックプロセスのデッドロックを当該マスタープロセスにより検出して自動的に再起動することができる。

【図面の簡単な説明】

【図1】 本発明の一実施形態を示すフローチャートである。

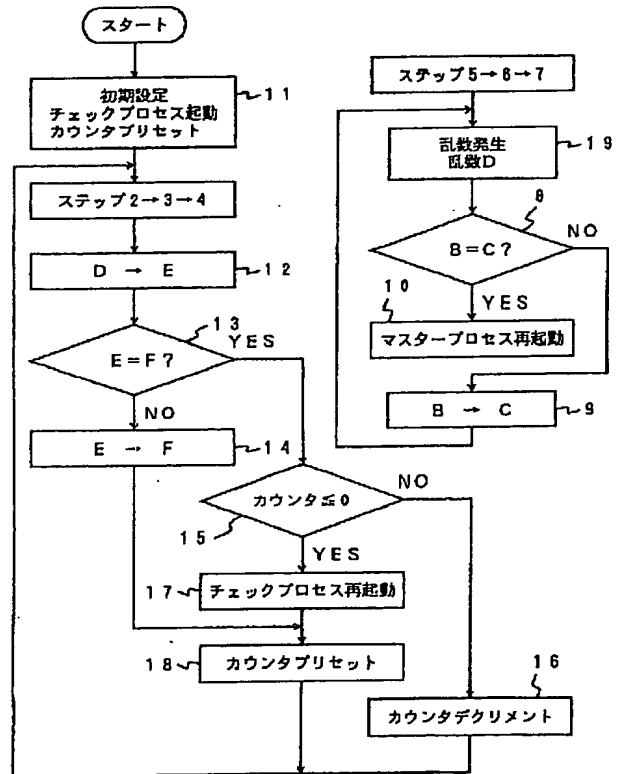
【図2】 本発明の他の実施形態を示すフローチャートである。

【図3】 従来の方法を示すフローチャートである。

【符号の説明】

- 1 初期設定、チェックプロセス起動ステップ
- 2 乱数書き込みステップ、乱数⇒A
- 4 イベント処理ステップ
- 5 チェックプロセス初期設定、周期設定ステップ
- 7 乱数ロードステップ、A⇒B
- 8 乱数比較ステップ、B=C?
- 9 乱数ロードステップ、B⇒C
- 10 マスタープロセス再起動ステップ

【図2】



【図 3】

